UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/477,688	01/06/2000	STEPHEN ANTHONY EDWARDS	1000/5	9589

35795 7590 03/11/2005

JONATHAN T. KAPLAN
ATTORNEY AT LAW
140 NASSAU STREET
NEW YORK, NY 10038-1501

EXAMINER	
ALI, SYED J	
ART UNIT	**PAPER NUMBER**
2127	

DATE MAILED: 03/11/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/477,688	**Applicant(s)** EDWARDS, STEPHEN ANTHONY	
	Examiner Syed J Ali	**Art Unit** 2127	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *04 February 2005*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-12* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-12* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on February 4, 2005 has been entered.

2. Claims 1-12 are presented for examination.

3. The text of those sections of Title 35, U.S. code not included in this office action can be found in a prior office action.

Claim Rejections - 35 USC § 112

4. **Claims 1, 3, and 10-12 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.**

5. In line of claim 1, lines 14-15 of claim 10, line 17 of claim 11, and line 11 of claim 12, there is reference to the context switch saving “a second state”. However, there is no mention of “a first state”. The actions relating to the “second state” imply that there is something done with a “first state”. However, this limitation is conspicuously absent from the claims.

6. In line 1 of claim 3, there is a lack of antecedent basis for the limitation of "the translation of the CCFG".

Claim Rejections - 35 USC § 103

7. **Claims 1-12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lin ("Efficient Compilation of Process-Based Concurrent Programs without Run-Time Scheduling") in view of Nilsen et al. (USPN 6,081,665) (hereinafter Nilsen).**

8. As per claim 1, Lin teaches the invention as claimed, including a method performed by a data processing system having a memory, comprising the steps of:

inputting a CCFG (pg. 213, Figs. 2[a-b], 3[a-b]);

scheduling the CCFG to produce a scheduled CCFG (pg. 215);

selecting a first node of the scheduled CCFG (pg. 213, Figs 2[a-c], elements p1, p2);

producing a first copy of the first node for an SCFG (pg. 213, Figs 2[a-c], elements p1, p2); and

coupling, if a first thread of the first node is suspended, between a second node of the SCFG of a second previously running thread and the first copy, a first context switch (Fig. 2[c], element c2).

9. Nilsen teaches the invention as claimed, including the following limitations not shown by Lin:

wherein the context switch saves a second state, of the second previously running thread, into a state variable dedicated to the second previously running thread (col. 37 lines 60-67).

10. It would have been obvious to one of ordinary skill in the art to combine Lin with Nilsen since the method of combining concurrent control flow graphs into sequential control flow graphs disclosed by Lin takes into account context switching between separate processes or threads, but fails to specify exactly how the code translation is generated, or what features are implemented therein. Rather, the preliminary ideas behind the construction of Petri net representations of sequential control flow graphs is disclosed, while the features related to scheduling and optimization are left up to the developer (pg. 215). The papers that Lin refers to presumably offer several scheduling techniques that can be used for the sequential control flow graph, yet the possible scheduling techniques are not limited to those references. Nilsen offers a technique for protecting the state of a thread during a context switch or a preemption by saving the information related to the suspended thread in a state variable and utilizing the information in the state variable upon resumption of the stopped thread, which would allow the thread and it's shared resources to remain in a consistent state during any context switches or synchronization.

11. As per claim 2, Nilsen teaches the invention as claimed, including the method of claim 1, wherein the first context switch is comprised of code that saves a state of a thread being suspended in a state variable (col. 37 lines 60-67).

12. Lin teaches the invention as claimed, including resuming another thread by performing a multiway branch on a state variable for a thread being resumed (pg. 213, Figs. 2[c] and 3[c]).

13. As per claim 3, Lin teaches the invention as claimed, including the method of claim 1, wherein the translation of the CCFG into the SCFG produces, for each node of the CCFG, at most one corresponding node in the SCFG (pg. 213).

14. As per claim 4, Lin does not specifically disclose the method of claim 1, wherein the step of scheduling further comprises a topological sort for determining the scheduled augmented CCFG.

15. "Official Notice" is taken that the use of topological sorts is well known and expected in the art, and would have been an obvious modification to Lin. Specifically, Lin teaches an ordered graph, with nodes used to refer to specific variable states and edges to traverse between those states (pg. 212-213, Figs. 2[a-c], 3[a-c]). A topological sort is well known as a way of ordering nodes, such that if a transition occurs between two nodes, then based on how that transition is represented, it is known what order the nodes occur in the graph. A common way of expressing this is that if an edge exists such that edge(u, v) is in the graph and u and v are nodes in the graph, then u comes before v in the ordering of the graph. This can be found in any number of programming guides, and an example is presented in the Boost Graph Library on the Boost C++ Library website (www.boost.org/libs/graph/doc/topological_sort.html). It would have been obvious to one of ordinary skill in the art to use a topological sort to determine the ordering of the scheduled augmented CCFG since the technique is well known, other programming methods that have been previously devised can be used in accordance with the sorting technique. Specifically, defining the graph with similar data structures would allow a programmer a multitude of predefined methods to operate on the data therein, depending on the

specific needs of each individual system. Furthermore, Lin makes mention of how various features of the sequential control flow graph must be mapped out in order to generate the necessary code. This is done via a pre-ordering method, which represents the flow of control in various data structures, including a representation of all the "reachable nodes". A topological sort would have been an obvious method of generating such a set of "reachable nodes" by defining all the nodes of the graph as well as the state transitions within a single data structure.

16. As per claim 5, Lin teaches the invention as claimed, including the method of claim 1, wherein an execution of the SCFG comprises translation of the SCFG into a programming language (pg. 213-216, §4.2).

17. As per claim 6, Lin teaches the invention as claimed, including the method of claim 5, wherein the programming language is C (pg. 213-216, §4.2).

18. As per claim 7, Lin teaches the invention as claimed, including the method of claim 1, further comprising a step of translation of the SCFG into a programming language (pg. 213-216, §4.2).

19. As per claim 8, Lin teaches the invention as claimed, including the method of claim 7, further comprising a step of executing the programming language translation of the SCFG (pg. 213-216, §4.2).

20. As per claim 9, Lin teaches the invention as claimed, including the method of claim 1, wherein an execution of the SCFG comprises interpretation of the SCFG (pg. 213-216, §4.2).

21. As per claim 10, Lin teaches the invention as claimed, including a data processing system having a memory and capable of implementing the method of claim 1 (Abstract).

22. As per claim 11, Lin teaches the invention as claimed, including a computer program product comprising a computer usable medium having computer readable code embodied thereon and capable of implementing the method of claim 1 (Abstract).

23. As per claim 12, Lin teaches the invention as claimed, including a computer data signal embodied in a carrier wave and representing sequences of instructions which, when executed by a processor, cause performance of the method of claim 1 (Abstract).

Response to Arguments

24. **Applicant's arguments filed February 4, 2005 have been fully considered but they are not persuasive.**

25. Applicant argues on page 2, "*Examiner is factually incorrect when he states that 'Lin takes into account context switching between separate processes or threads.*'"

26. Examiner respectfully disagrees. Lin is directed to synthesizing disjoint processes or threads that have data interdependencies into a single sequential process or thread. For example,

Figs. 1 and 2 contemplate switching between two processes, "ping" and "pong" upon receiving a particular input at a node of the graph. When the flow of execution switches from one process or thread to another, a context switch occurs. Applicant's argument diverges into a discussion of how Lin's method of generating a sequential flow of execution does not meet the claim limitations because Lin allegedly duplicates subgraphs. While this argument is addressed more fully below, it is not apparent how the alleged duplication of subgraphs fails to show a context switch. A context switch, by definition, is when a multitasking operating system stops running one process and starts running another (see the Free On-Line Dictionary of Computing at http://wombat.doc.ic.ac.uk/foldoc). This is exactly what is achieved by the parallel composition disclosed by Lin. At particular transition points, e.g. c1 and c2 of Fig. 2, control either shifts to another process or remains within the original process. This, by definition, is a context switch.

27. Applicant presents an example of context switching that allegedly sets the claimed invention apart from Lin. Applicant argues that "*Lin uses subgraph duplication, rather than context switching,*" which cannot generate a sequential control flow graph "*that can eliminate the subgraph duplication*" of the specific example presented.

28. The example of context switching discussed by Applicant, while supported by the claims, is not the only type of sequential flow graph that can be generated from a concurrent control flow graph. Applicant's Fig. 7 addresses two disjoint processes or threads that do not necessarily have any data dependencies between them. As the claims are presented, a context switch may occur between two disjoint processes or two processes that are interrelated. In such a circumstance, Lin would not duplicate subgraphs since the common transitions are collapsed.

Essentially, the synthesis of a sequential flow of control seeks to internalize communication points into "synchronization points" that allow the flow of execution to switch between processes. Nonetheless, there is no requirement in the claims, explicit or implied, that duplication of subgraphs is not permissible. The claimed invention takes a concurrent control flow graph and generates a sequential flow of control, while switching between execution contexts. Lin takes control flow graphs of two concurrent processes that have data interdependencies, and synthesizes a sequential flow of control. This is achieved by internalizing the data dependencies at points where the flow of execution switches between processes.

Conclusion

29. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Syed J Ali whose telephone number is (571) 272-3769. The examiner can normally be reached on Mon-Fri 8-5:30, 2nd Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai T An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Syed Ali
February 28, 2005

MENG-AL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100